

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION ESDS Reuse Working Group

Software Reuse Practices Within the Earth Science Community

Abstract Reference Number: 9081

Paper Number: IN23B-1219

Authors: Olding, S W, Delnore, V E, Samadi, S, Wolfe, R E

The Earth Sciences software development community is often challenged to provide cost effective, highly reliable and easy-to-use software to achieve scientific missions. In the process, the Earth science community spends a significant amount of resources developing software components and other software development artifacts that may also be of value if reused in other projects requiring similar functionality. Indeed, the software engineering literature cites many case studies where reusing existing software artifacts has improved productivity and quality while reducing system development cost and schedule. However, realizing such benefits for Earth science data systems has been challenging due to the scale, complexity, heterogeneity and distributed nature of these systems, which often are constructed and operated by a mix of government, industry and academic organizations. Although new generations of the more complex systems often exploit domain knowledge and expertise from previous development activities, a more disciplined reuse approach is still needed to further assist with cost reduction and productivity improvement within the Earth science community. A recent survey performed by the NASA Earth Science Data Systems Reuse Working Group examined current reuse practices within the Earth science community and community opinions on various approaches to facilitating software reuse. In this presentation, we present the findings of the study and put forward suggestions for increasing reuse and improving collaboration within the Earth science software development community.

Expected Benefits of Reuse

- Lower development costs
- Higher productivity; better use of resources
- Reduce cycle time; quicker development
- Lower training costs
- Easier maintenance
- Higher quality
- Lower risk
- Better interoperability

The 'application specific' layer represents software which is unique to a particular application. This is not software that can be reused.

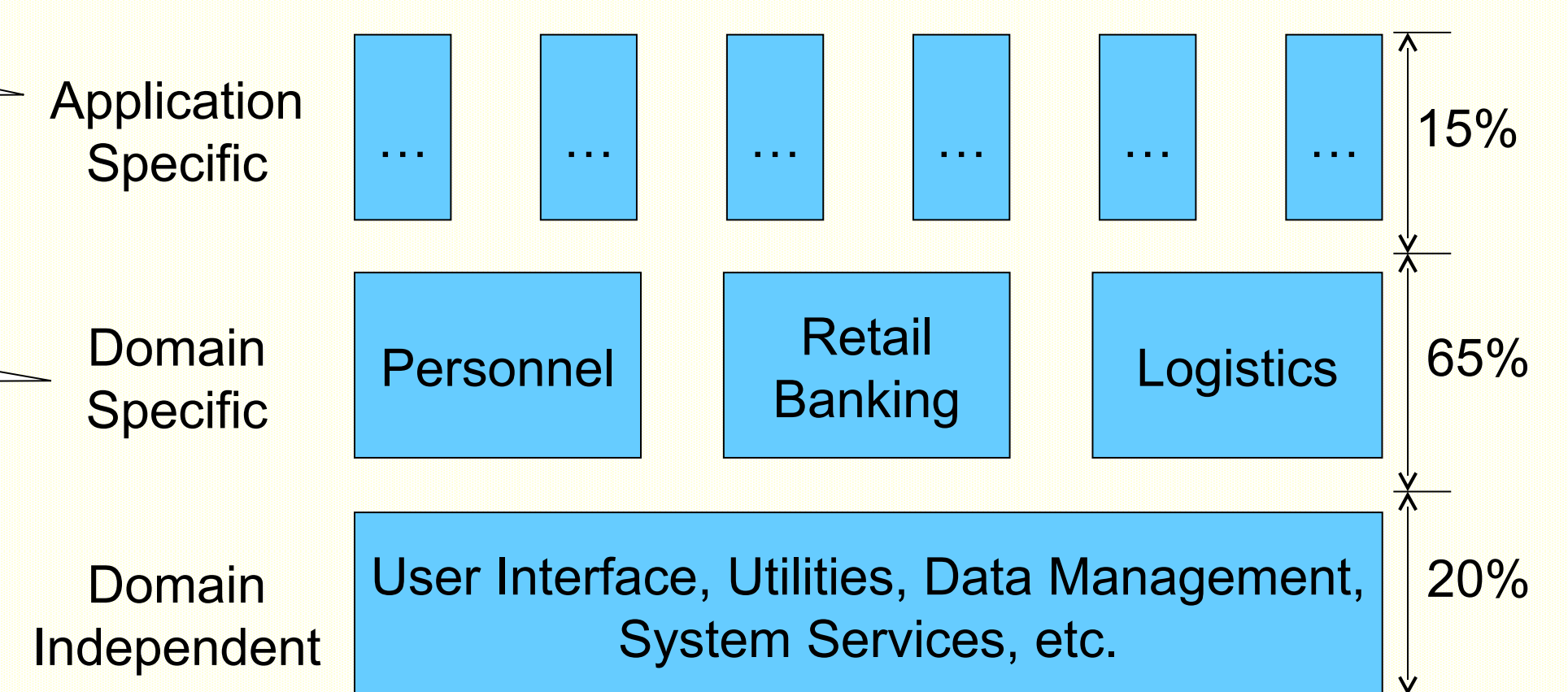
The 'domain specific' layer represents software that is specific to a particular business area, e.g. a Earth science missions. Software in this layer has the potential to be reused across similar types of systems.

Reuse in the 'domain independent' layer usually comes from commercial off-the-shelf (COTS) components and public open source software.

Software reuse is the process of implementing or updating software systems using pre-existing software assets

Reusable assets can be from any part of the software development life cycle including: software components, objects, software requirement analysis and design models, domain architectures, database schemas, code documentation, test scenarios, and plans

The composition of a 'typical' application...



... a theoretical reuse potential of up to 85% of new application development.

Source: Jeffery S. Poulin, "Measuring Software Reuse"

Survey conducted to establish reuse practices and needs of the Earth science community

OMB approval obtained 01/04/2005 (Approval No.: 2700-0117)

Used Web-based survey tool (Vanguard Vista) to simplify response submission and data collection

Approximately 3000 invitations issued -100 responses

ESDS Software Reuse Questionnaire

Over the last five years, how often have you (or your project) reused the following types of software development artifacts?

	1 (Never)	2 (Rarely)	3 (Sometimes)	4 (Often)	5 (Very Often)
a) Algorithms, Techniques	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) Designs, Architectures	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) Source code, Scripts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) Executables, Binaries	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) Other	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

If Other, please specify what other types of software development artifacts you have reused:

Over the last five years, how often have you (or your project) reused the following types of software development artifacts?

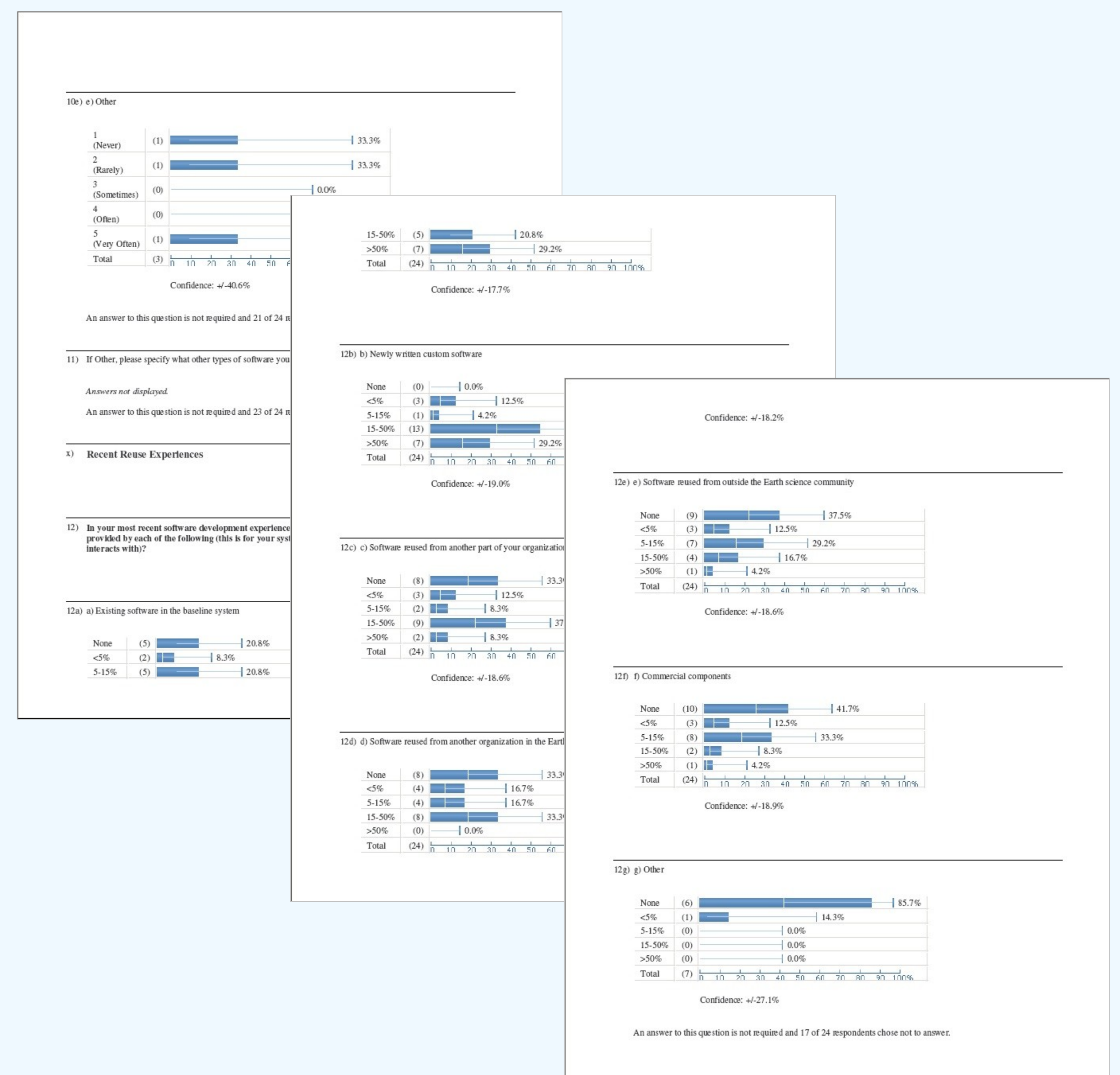
	1 (Never)	2 (Rarely)	3 (Sometimes)	4 (Often)	5 (Very Often)
a) Complete systems or applications	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) Subsystems or components	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) Code libraries	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) Code fragments	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) Other	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

If Other, please specify what other types of software development artifacts you have reused:

Recent Reuse Experiences

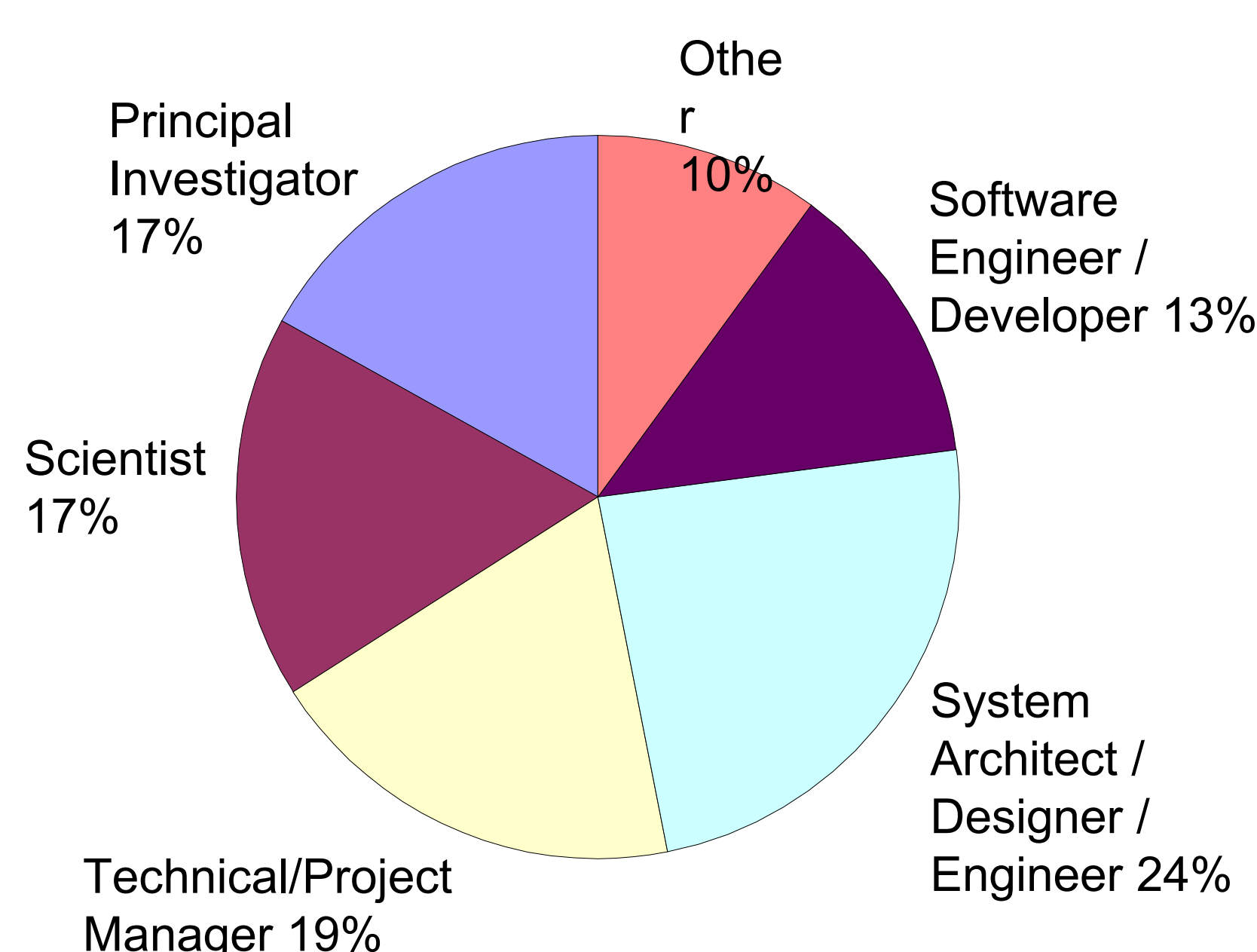
For any of the newly written software, was there software from another source that might have provided "any" of the capabilities needed? If so, how important were the following factors in preventing you from reusing that software?

	1 (Not important at all)	2 (Not very important)	3 (Somewhat important)	4 (Important)	5 (Very important)
a) I didn't know other software existed at the time	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) Other software wasn't compatible with my system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) Other software didn't exactly match my requirements	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) Other software was difficult to understand or poorly documented	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) Other software was too complex or difficult to adapt to my needs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
f) It was hard to overcome licensing restrictions	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
g) I needed the source code and it wasn't available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
h) I preferred to have the development take place within my project or wanted the experience of developing the needed capability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
i) I didn't like how the other software was designed/implemented	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
j) Other	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

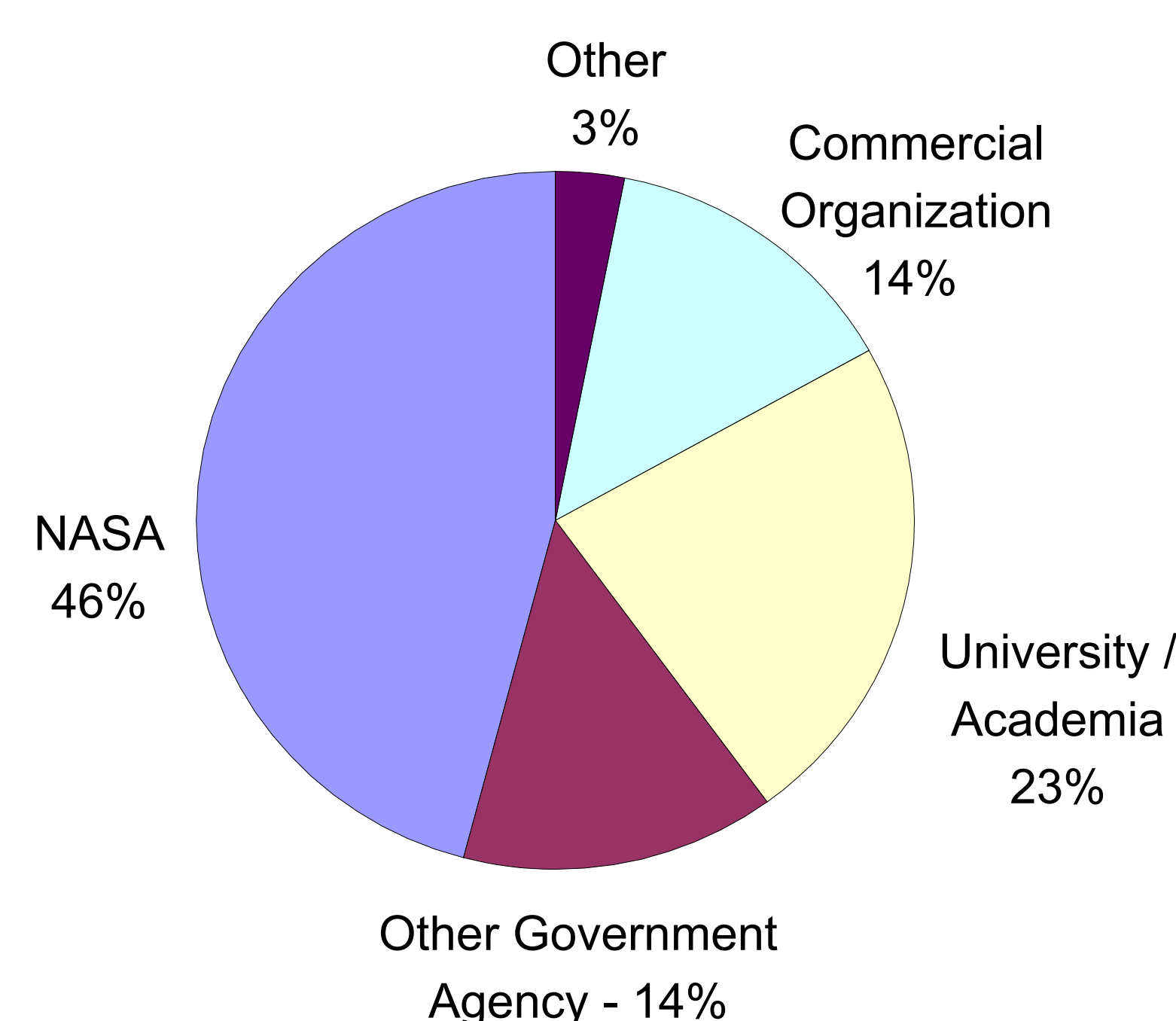


Reuse Survey – Community Profile

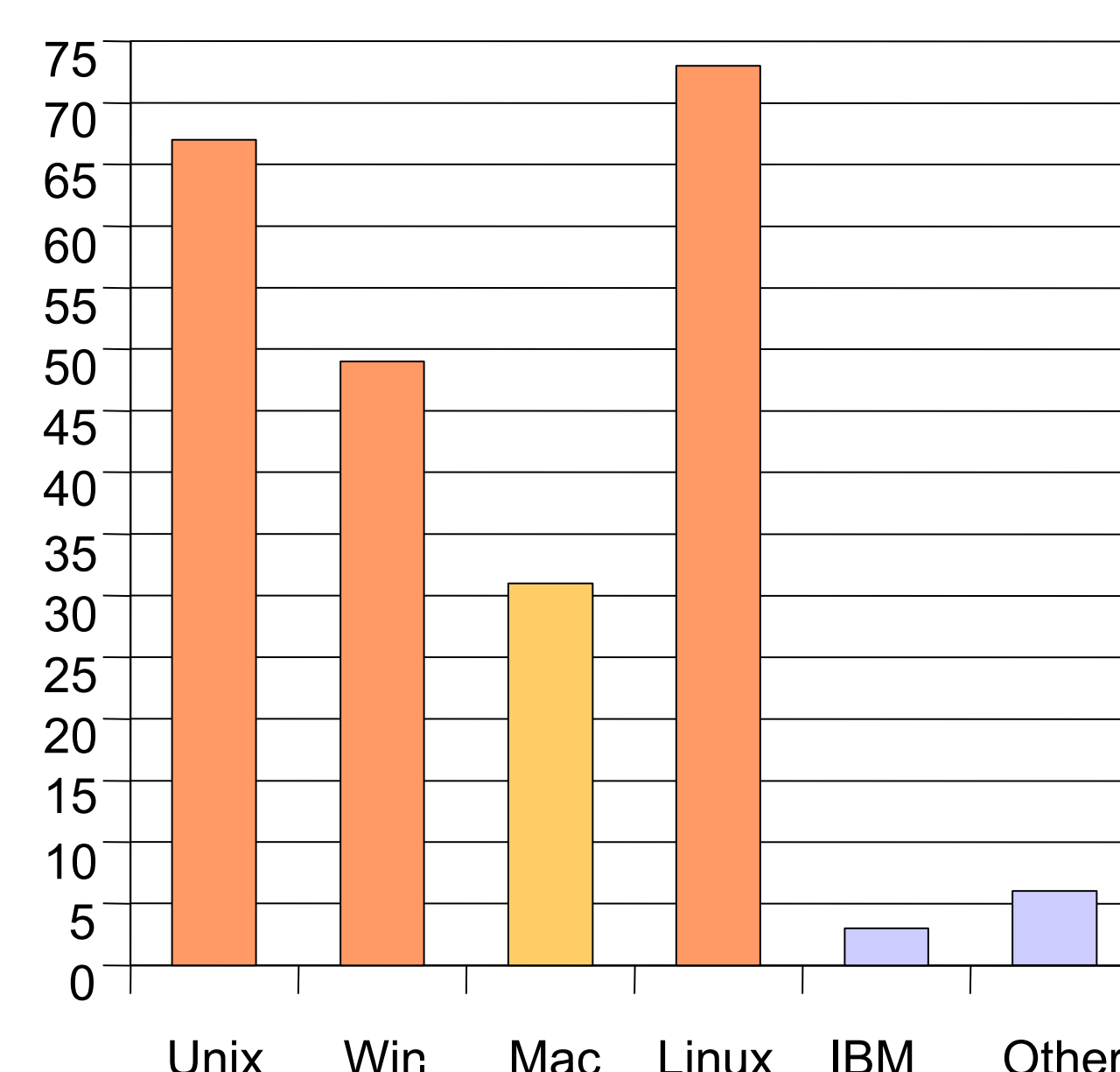
Which of the following best describes your main role in your project/software development process?



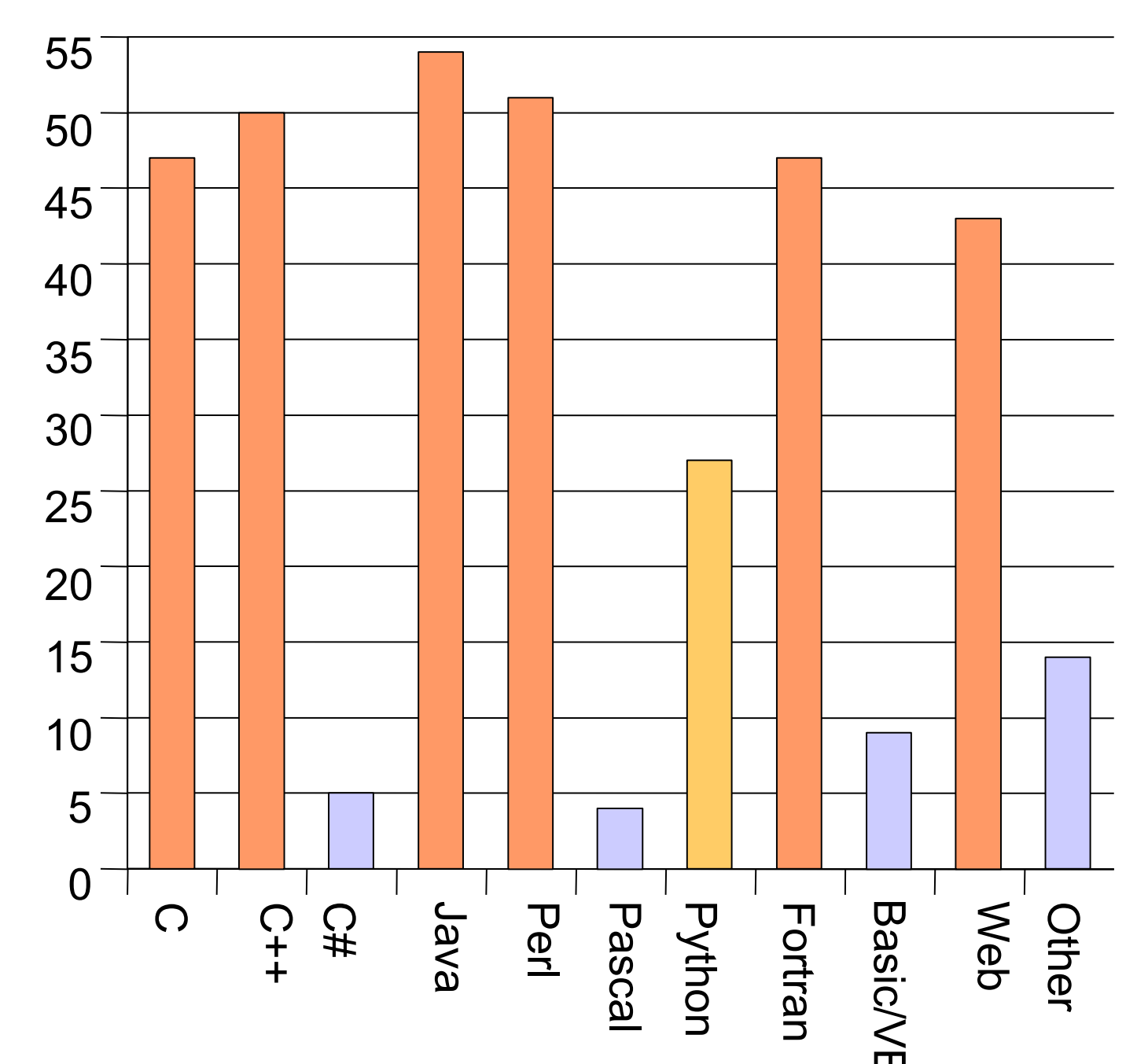
Which category best describes the type of your organization?



Which Operating System(s) do you currently use or plan to use... ?

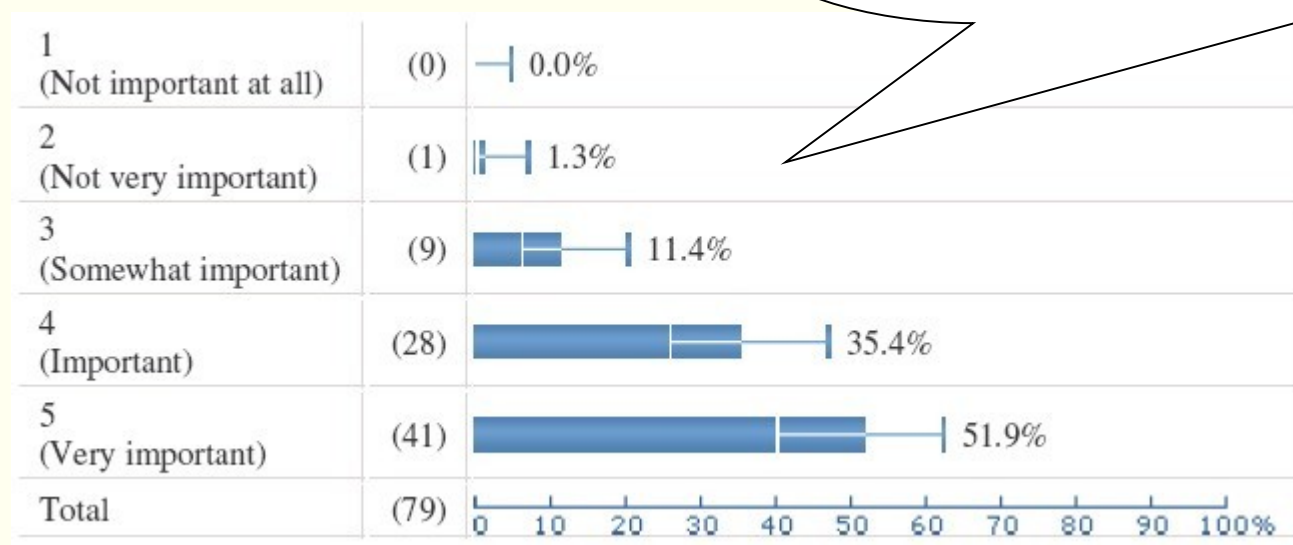


Which programming language(s) do you currently use or plan to use... ?



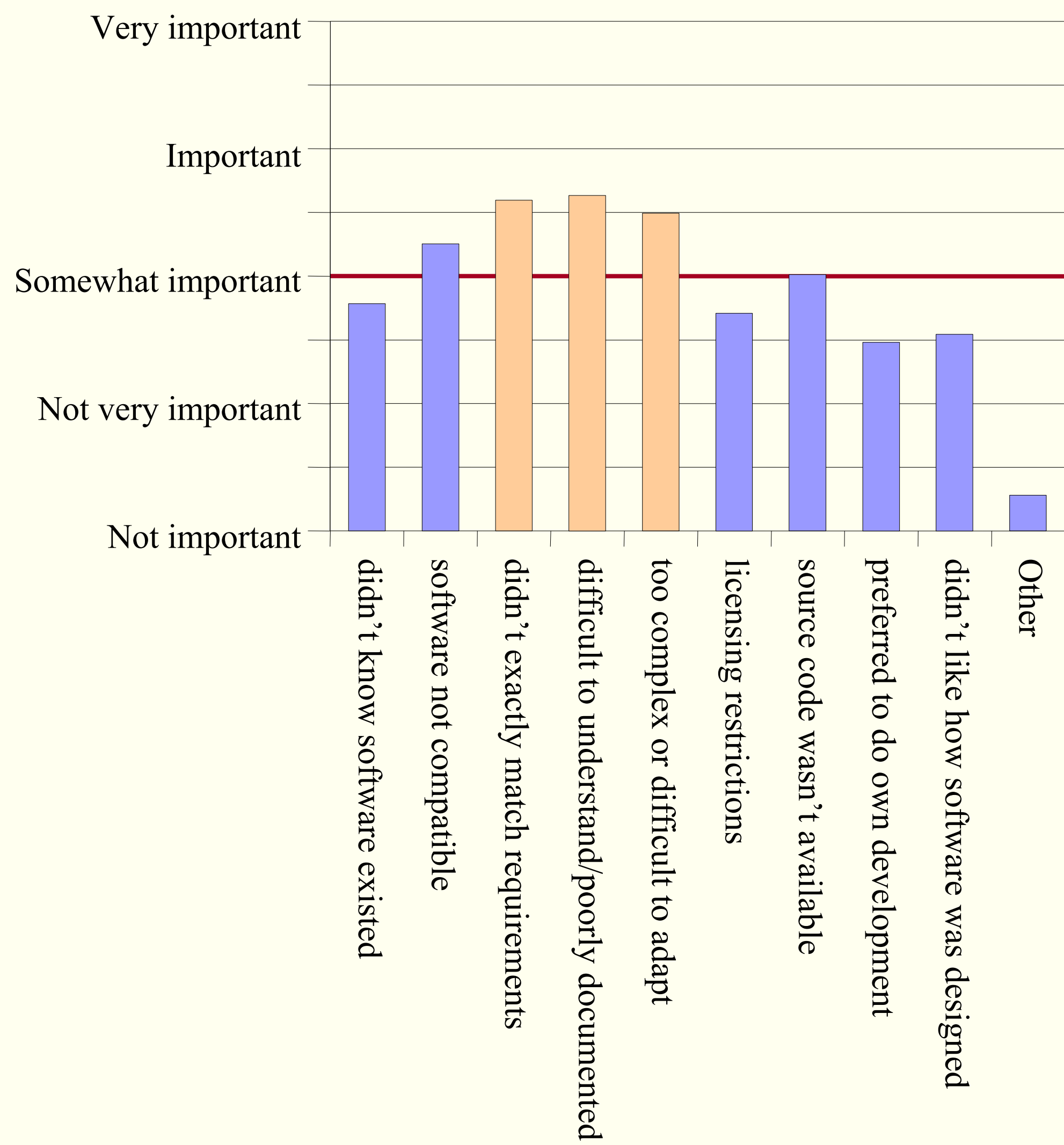
Survey Results and Conclusions

I. Reuse Experiences

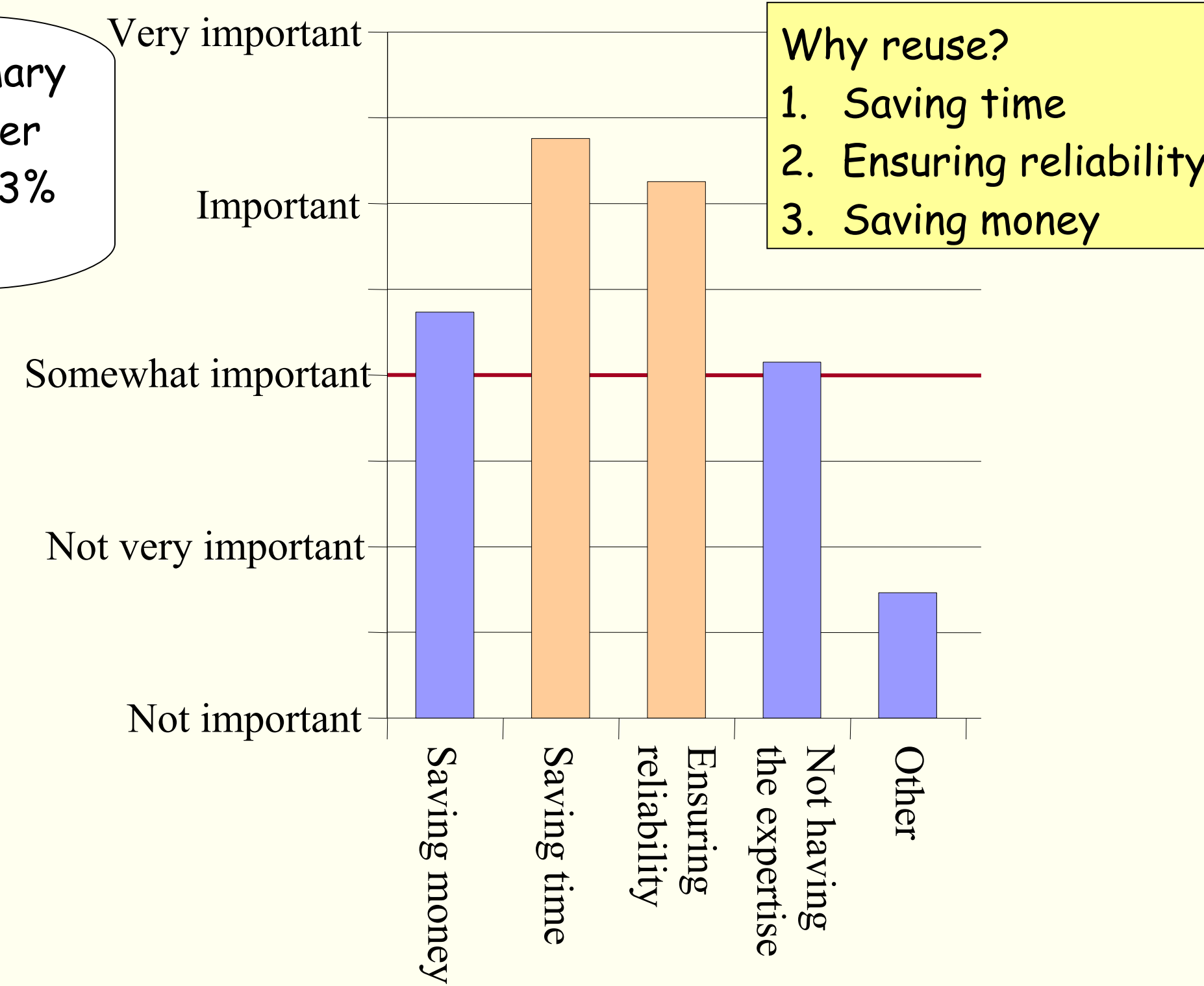


Saving time was ranked as the primary motivation for reuse. Rated as either important or very important by 87.3% of respondents

Barriers to Reusing Software

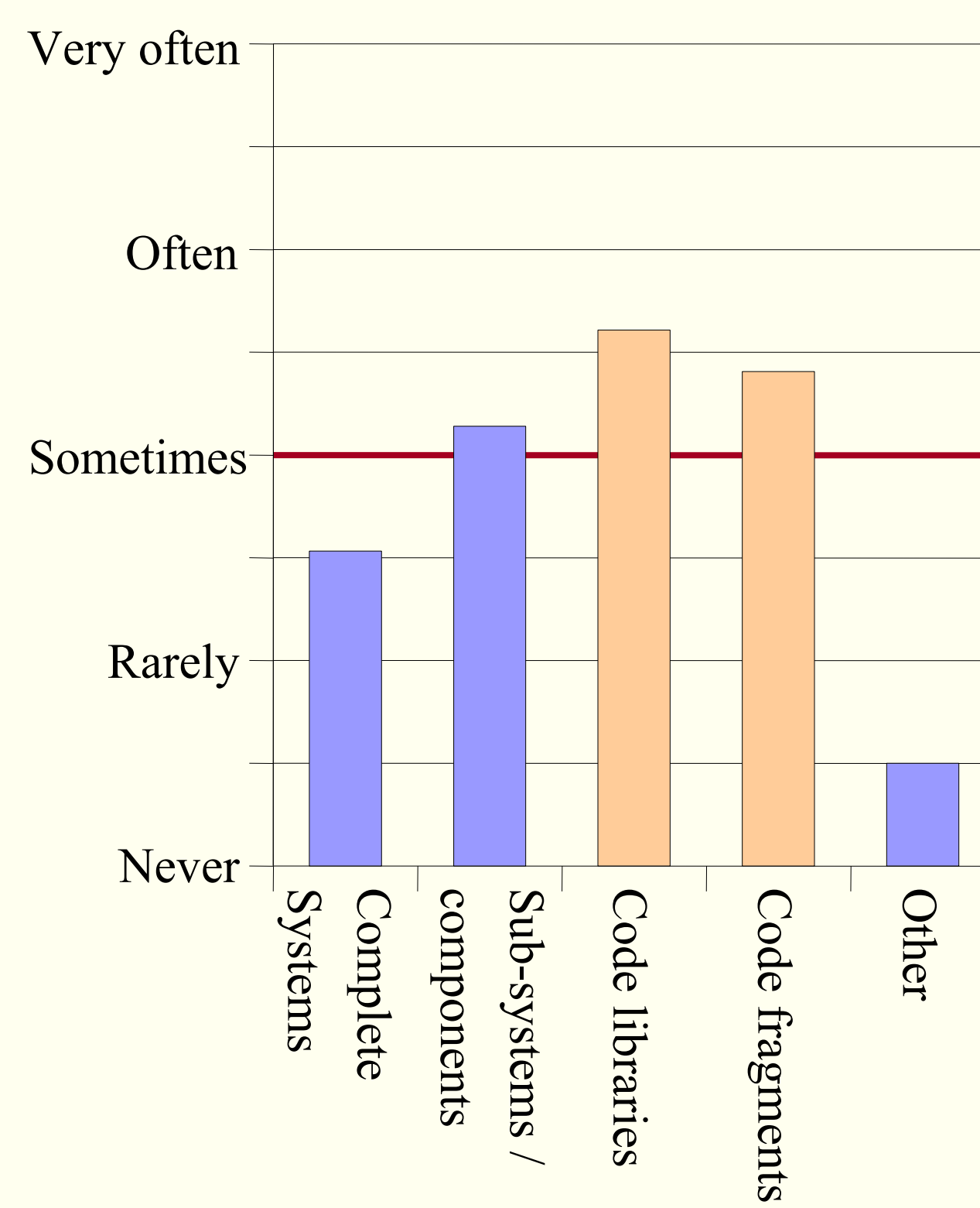


Motivation For Reuse



Why reuse?
1. Saving time
2. Ensuring reliability
3. Saving money

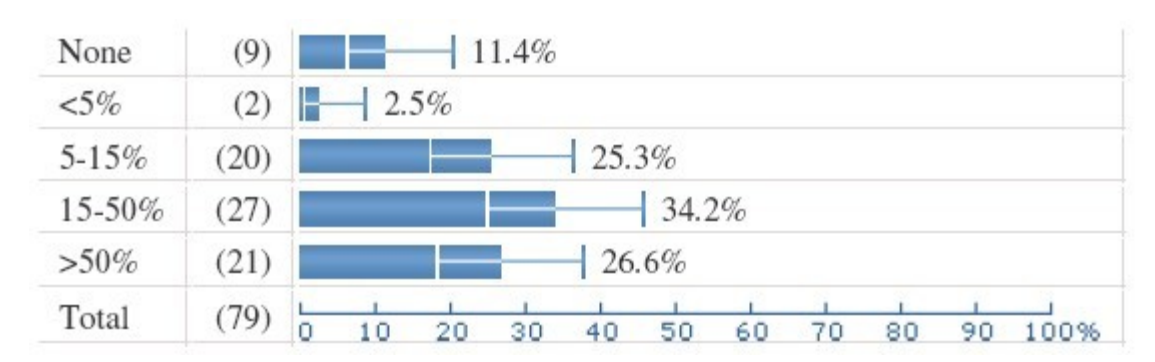
Types of Software Reused



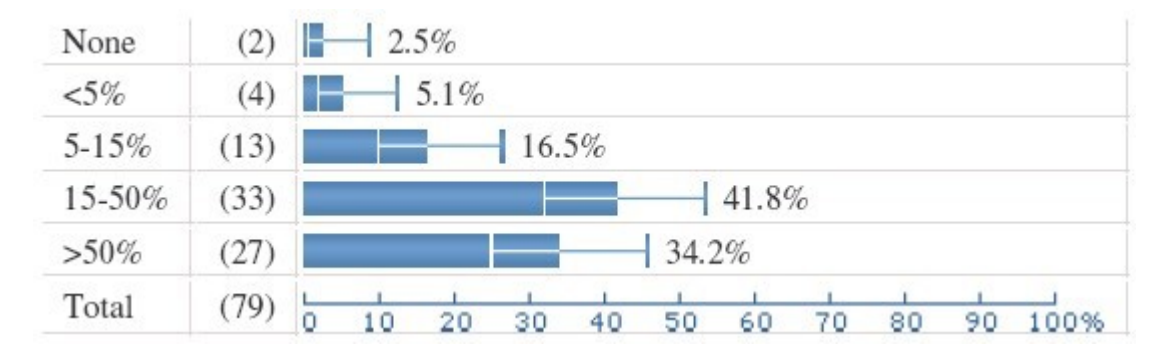
Reusers tend to favor the smaller units of code while the software that is made available for reuse is more often the larger units of code

II. Software Functionality Sources

Existing software in the baseline system



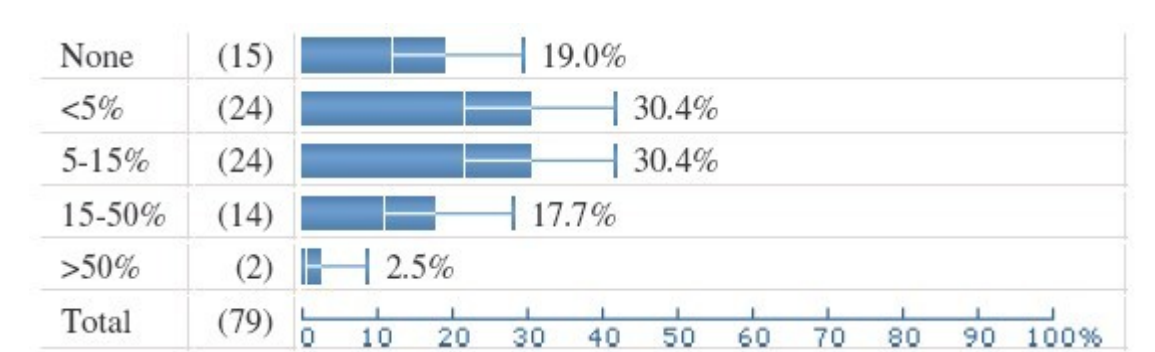
Newly written custom software



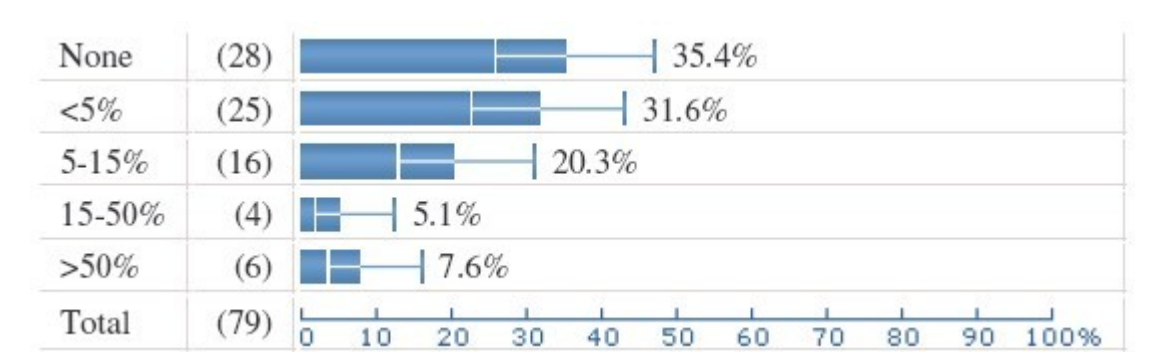
Software reused from another part of your organization



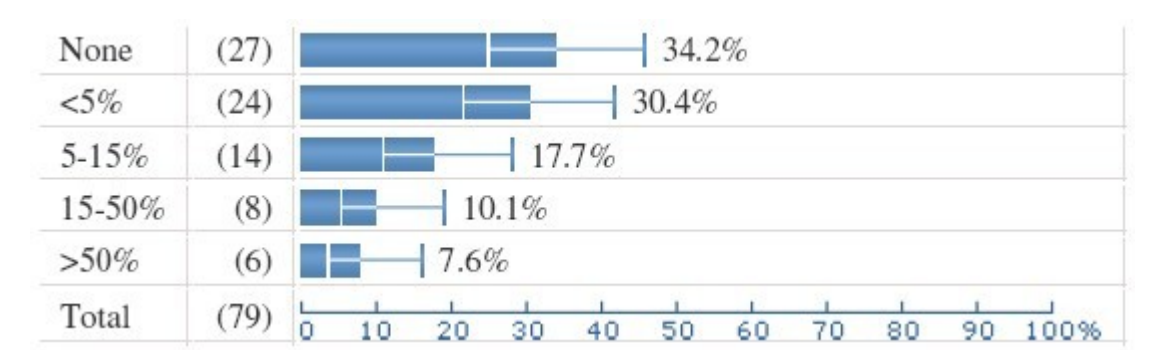
Software reused from another organization in the Earth science community



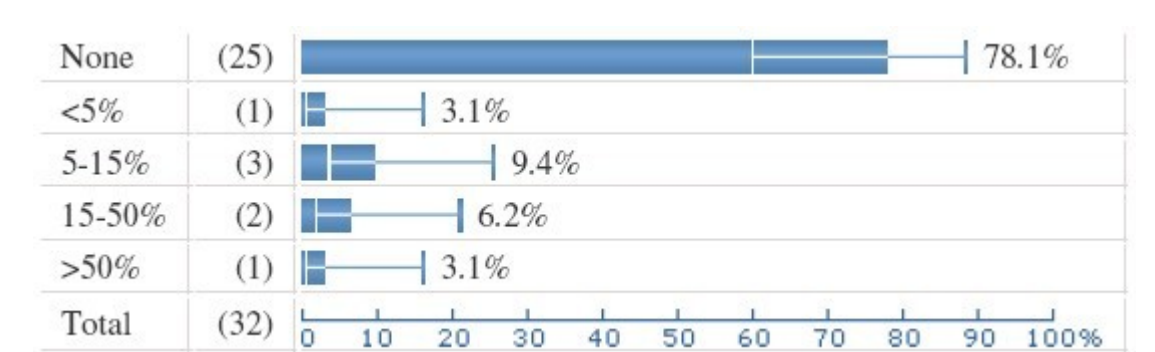
Software reused from outside the Earth science community



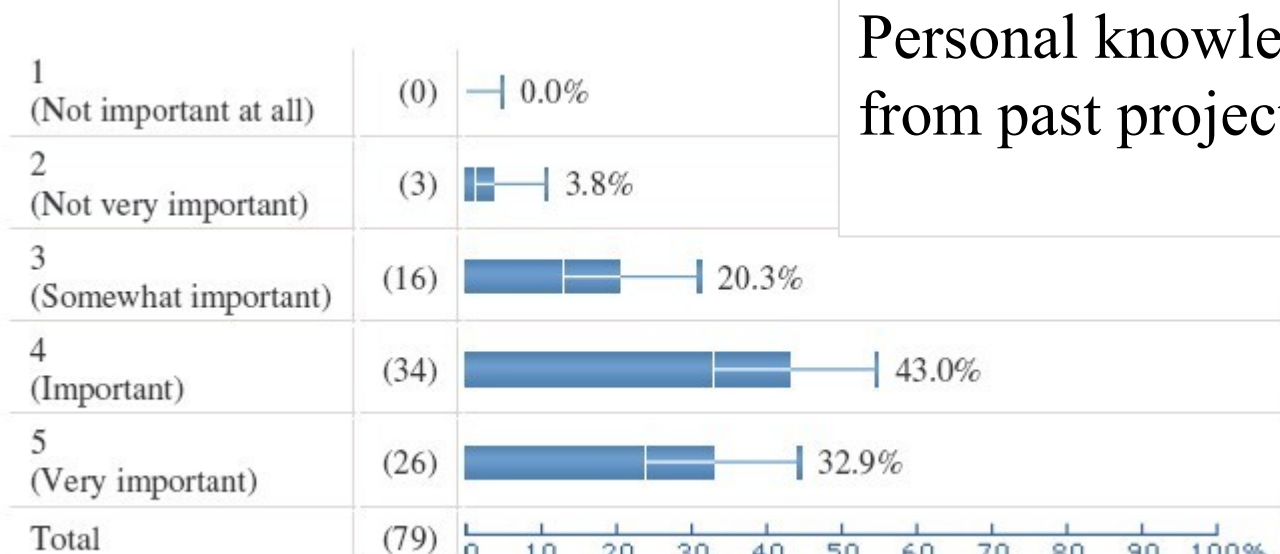
Commercial components



Other



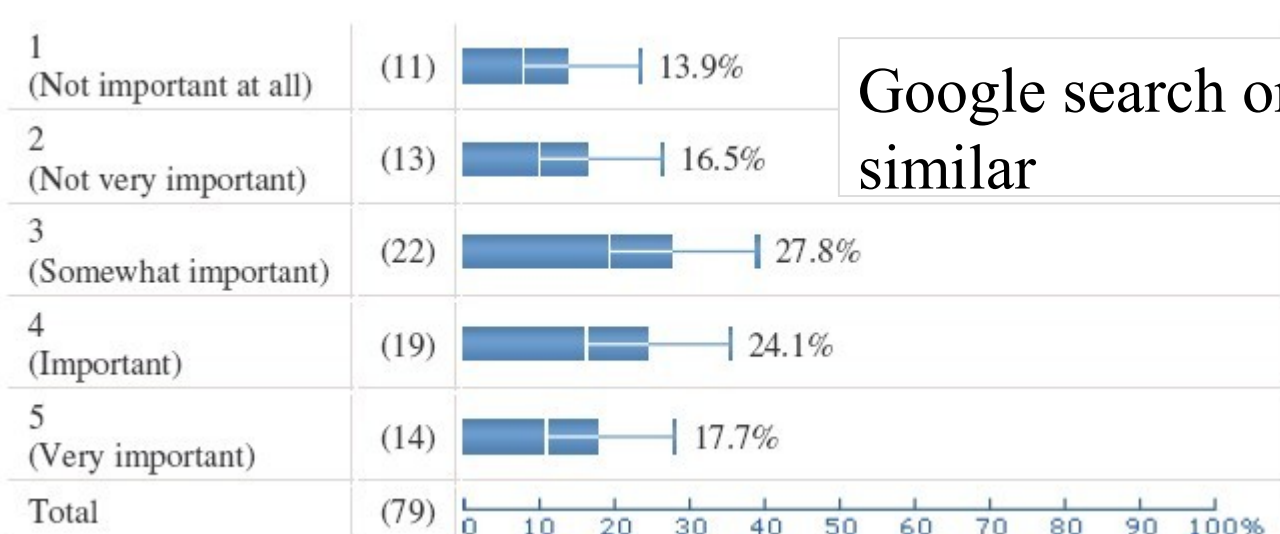
III. Locating reusable software development artifacts



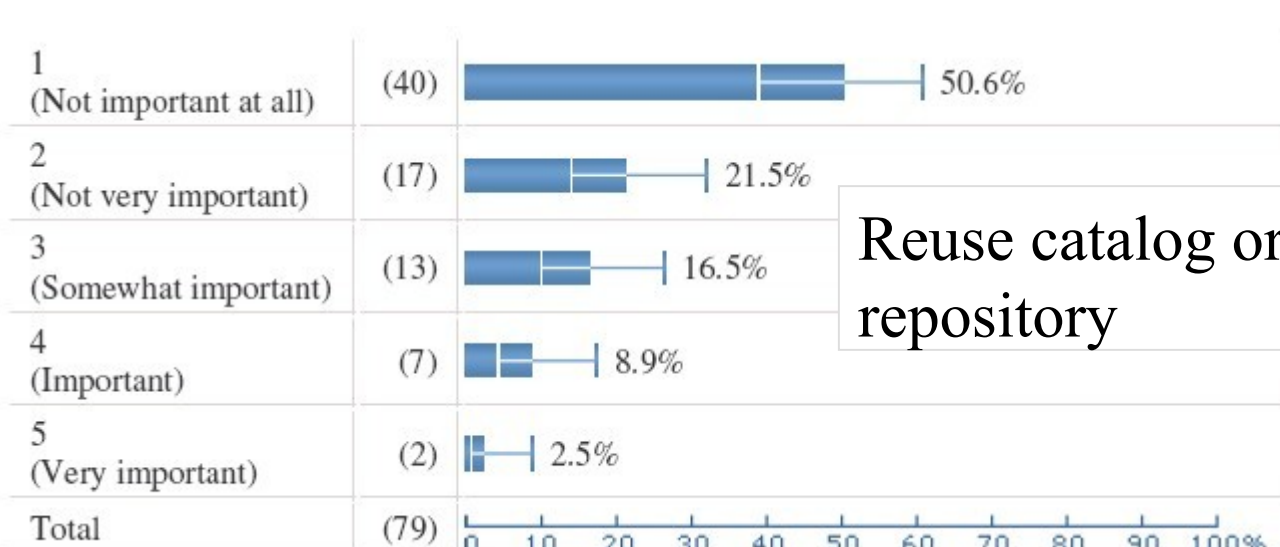
Personal knowledge from past projects



Word of mouth/networking



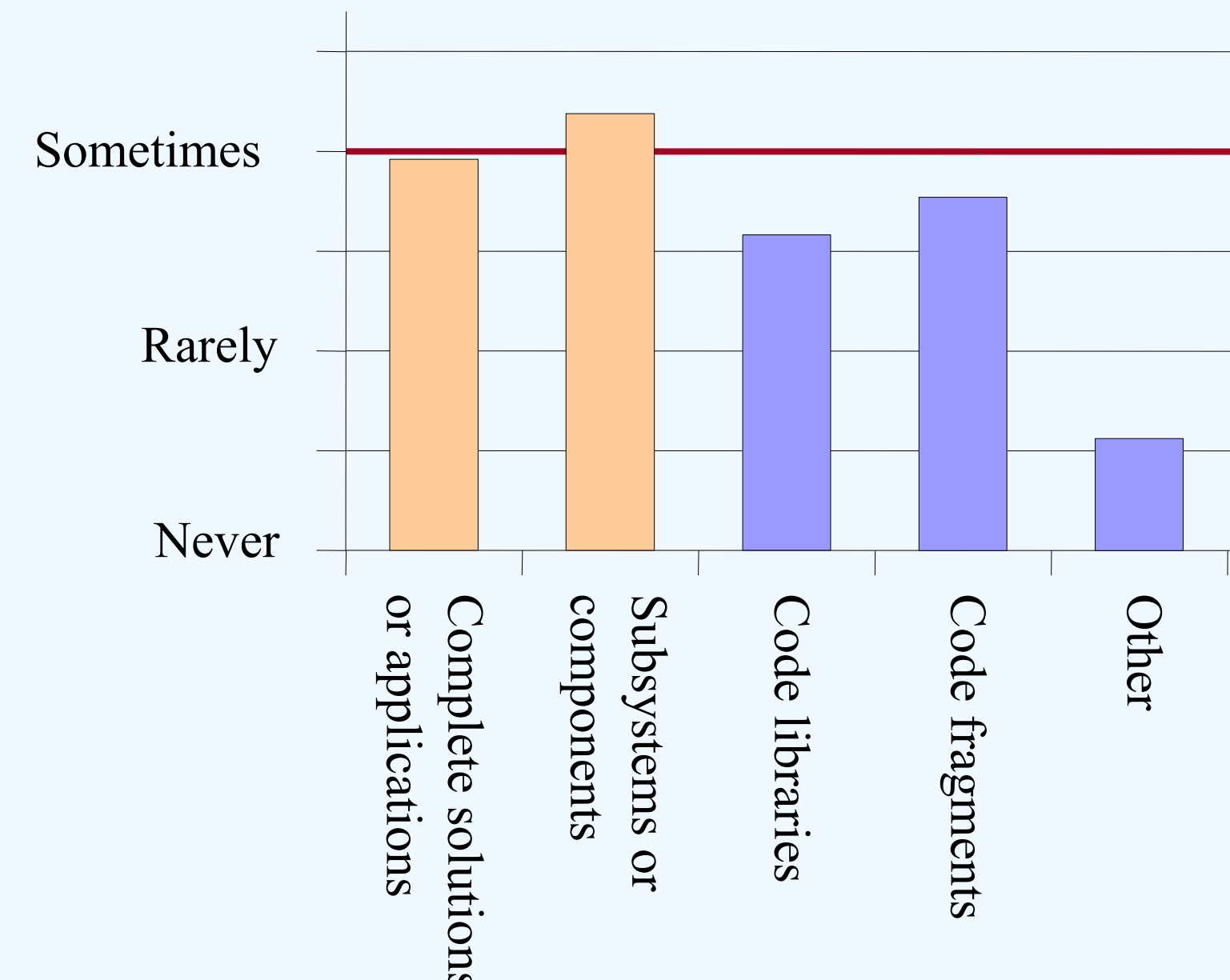
Google search or similar



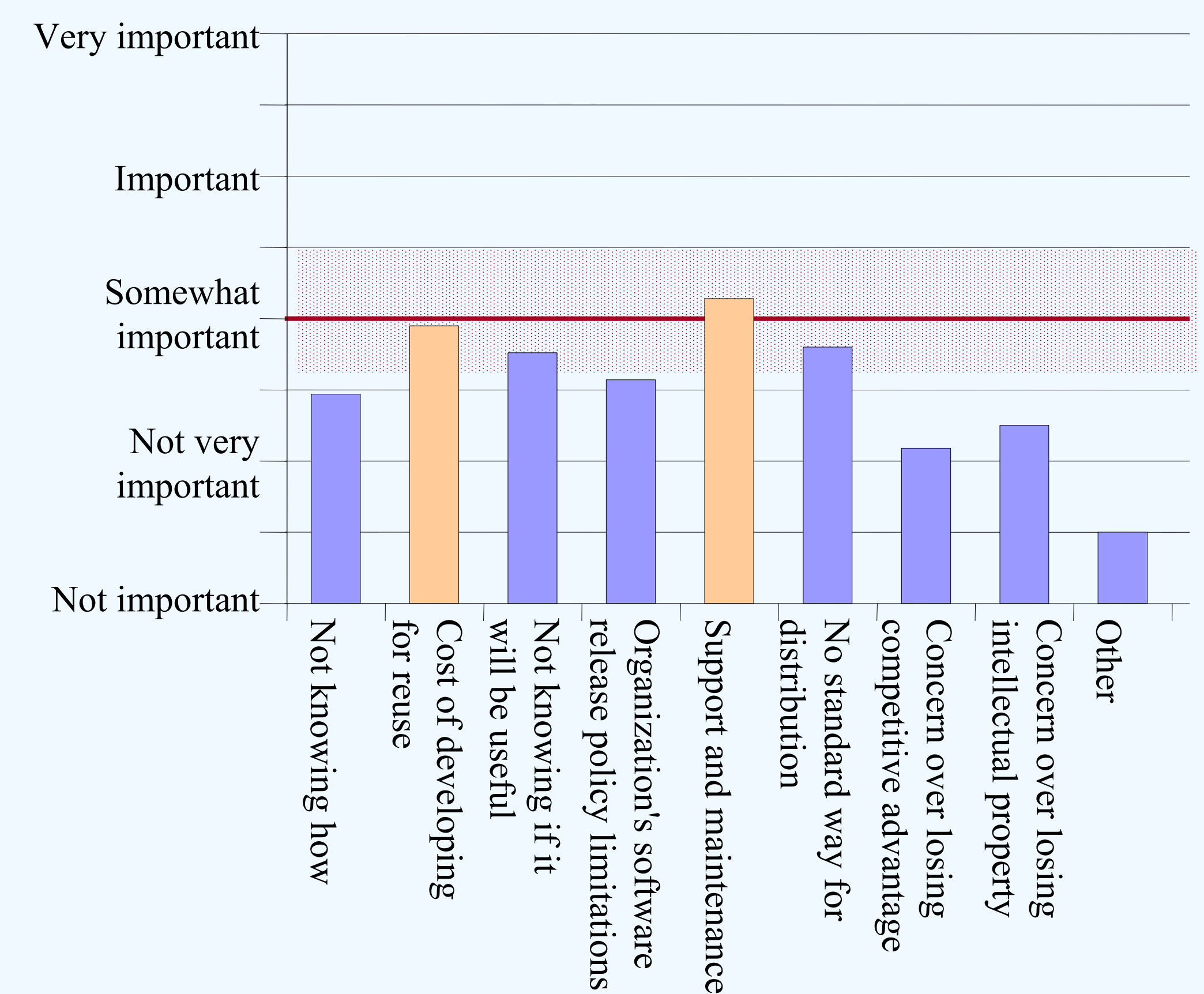
Reuse catalog or repository

IV. Creating reusable software

Types of Software Created for Reuse

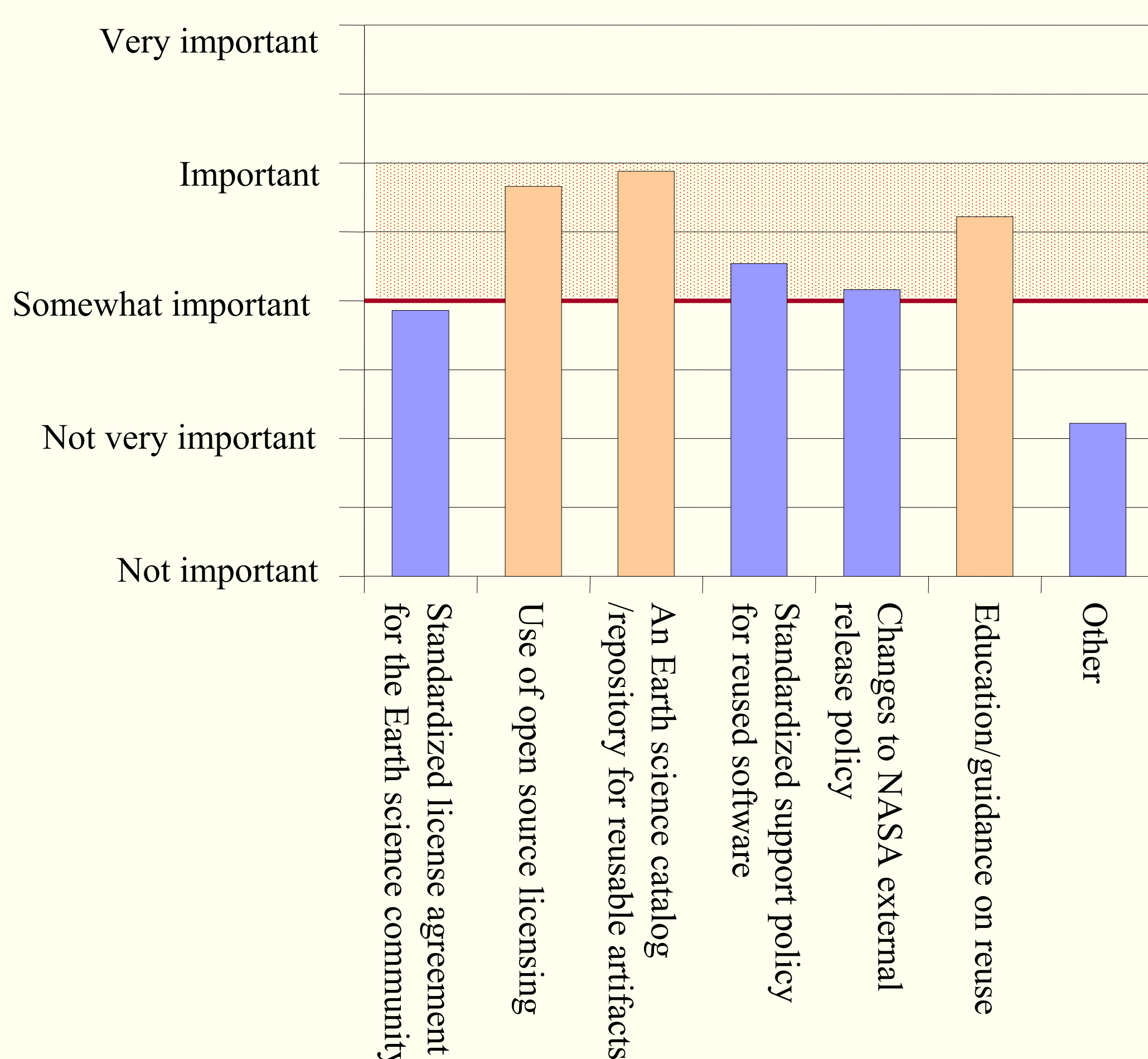


Barriers to Making Software Available for Reuse



V. Increasing Software Reuse in the Earth Science Community

How to Increase Reuse



Greater use of open source licensing, a repository for Earth science artifacts, and education and guidance on reuse were identified as the most important factors in increasing the amount of reuse.

The Earth Science Data Systems Reuse Working Group has established a pilot reuse portal to:

- Raise awareness of software reuse within the Earth science community,
- Establish a platform for community members to share/exchange resources with each other,
- Be the gateway for reuse information relevant to the community,
- Make access to reuse resources easier, and
- Become the major starting site for reuse within the community



<http://softwarereuse.nasa.gov/>